

作业六

7.1-4

要实现非递增排序，只需要将判断条件从 \leq 修改为 \geq ，伪代码如下。

```
FUNCTION QuickSortDesc(arr, low, high)
    IF low < high THEN
        pivotIndex ← PartitionDesc(arr, low, high)
        QuickSortDesc(arr, low, pivotIndex - 1) // 排序左子数组
        QuickSortDesc(arr, pivotIndex + 1, high) // 排序右子数组
    END IF
END FUNCTION

FUNCTION PartitionDesc(arr, low, high)
    pivot ← arr[high] // 选择最后一个元素作为枢轴
    i ← low - 1       // 指向大于枢轴区域的最后一个元素

    FOR j ← low TO high - 1 DO
        IF arr[j] ≥ pivot THEN
            i ← i + 1
            SWAP arr[i] WITH arr[j] // 将大于等于枢轴的元素交换到前面
        END IF
    END FOR
    SWAP arr[i + 1] WITH arr[high] // 将枢轴放置到正确位置
    RETURN i + 1 // 返回枢轴的位置
END FUNCTION
```

7.4-2

快速排序的时间复杂度由每次**划分**的操作 $O(n)$ 和划分后对子数组递归调用快速排序。

设快速排序对大小为 n 的数组所需时间为 $T(n)$ 。在每次划分时：枢轴选定后，将数组分为两部分，大小分别为 k 和 $n-k-1$ ，递归关系可表示为 $T(n) = T(k) + T(n - k - 1) + O(n)$

在理想情况下，数组被均匀划分为大小近似相等的两部分 $k = \frac{n}{2}$

递归公式变为 $T(n) = 2T\left(\frac{n}{2}\right) + O(n)$

根据主定理可知总时间复杂度为 $T(n) = O(n \log n)$